Audio on Linux: End of a Golden Age?

Lars-Peter Clausen – Analog Devices
Agenda

• History
  – Major transitions in software and hardware architecture

• Present
  – A look at the current situation
  – Are we in a golden age?

• Future
  – What major transitions lie ahead of us
  – How are we going to react to them?
Interdependent vs. Modular
- No clear boundaries defined between sub-modules
- Different sub-modules are aware of each other's internals
  - Creates dependencies
- Parts can't be upgraded or modified independently of each other
Modular

- Partitioning in sub-modules
- Clearly defined functions and interfaces
- Parts can be changed independently of each other
  - Drop-in replacements
- Constraint by the interface
History
Humble Beginnings
PC Speaker (Beeper)

- Found in all IBM compatible PCs
  - Present in the first IBM PC 5150 (1981)
- Has only two states
  - Toggling a specific frequency generates a tone (PWM)
- Magnetic or Piezoelectric plate
- In Linux supported by the input framework
Extending Features
• First widespread consumer sound card
  – Soundblaster 1.0 release in 1989
• Primarily synthesizer based
• Mono PCM channel
• Became defacto standard for consumer sound cards
  – Many applications expected a sound blaster interface
  – Other manufacturers included a Soundblaster compatibility mode in their hardware
Audio on Linux
Open Sound System (OSS)
Open Sound System (OSS)

- Used to be default audio subsystem in v2.4
- /dev/dsp interface
  - To playback audio use `write()`
  - To capture audio use `read()`
  - Some IOCTLs for management task
- Limited to one soundcard per system
Open Sound System (OSS)

- Linux today still supports OSS
  - Natively – A few OSS only drivers remain
  - Emulation – Through ALSA
- Disabled in all major distributions today
- Userspace emulation is available
Advance Linux Sound Architecture (ALSA)
ALSA

- First release in 1998
- Was added to the kernel during v2.5 development in 2002
- Replaced OSS
ALSA – Basic Architecture

- Split into kernel and userspace components
- Accurately describe the hardware capabilities
  - No emulation of missing hardware functionality at the lowest level
- Server/Client architecture
- Modular plugin architecture
  - Stackable modules
  - Implement emulation in modules
ALSA – Basic Architecture

- Organized into
  - Sound card (physical hardware device)
  - Device (PCM, mixer, MIDI, timer)
  - Subdevice (Specific endpoint)
ALSA – Controls

- Allows to control device configuration
- Different types
  - Volume (with gain table)
  - Switch
  - Enumeration
- Each control has a name
  - Follows standard naming scheme
ALSA – Constraint System

- Each PCM stream has a configuration space
  - 14 different parameters
- Used to negotiate stream configuration between userspace and driver
- Allows to progressively discover the capabilities of the hardware

<table>
<thead>
<tr>
<th></th>
<th>Rate</th>
<th>Ch 1</th>
<th>Ch 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>16k</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>32k</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Reducing Cost
Software Soundcards

- CPUs became a lot more powerful in the mid-90s
- The relative amount of CPU usage for audio processing became small
- Hardware is simplified
  - No more synthesizers
  - No more mixers
- Features are moved into software
Audio CODEC '97 (AC'97)

- Standardization of audio devices
- Introduced in 1997
- Split into controller and CODEC
- Common control and data bus
- CODEC standard register map
  - Discoverable feature set
  - Vendor specific extensions
USB Audio Class

- USB defines standard device classes
  - Compatibility between different devices offering the same service
- USB Audio Class was part of the first USB standard (1.1) in 1998
- ALSA added USB audio class support in 2002
The Sound Server Wars
The Sound Server Wars

- Users want to be able to play multiple audio streams at the same time
- Hardware no longer has mixing capabilities
- Introduction of sound servers
  - Mix multiple streams in software
  - Forward mixed stream to ALSA
The Sound Server Wars

• Different Desktop environments adopted different sound servers
  – artsd, KDE project
  – ESD, Enlightened Desktop Environment and GNOME

• Each sound server had their own client API
  – Applications had to choose which API to use

• It is not possible to use applications using different APIs at the same time
PulseAudio

- Development started in 2004
- Distributions started shipping it in 2007-2008 as the default sound server
- Provided compatibility layers for other APIs
  - Won the sound server wars
- Simplified audio API
  - Good for application adoption
PulseAudio – A Modern Sound Server

• Introduced many differentiating features
  – Timer-based audio scheduling
    • Low latency and power-saving
  – Per application volume
  – Network capable
  – Bluetooth integration
  – Multi-user capable
• Has virtually replaced all other sound servers
  – Default on all major distributions
Embedded on the Rise
ALSA for System on a Chip (ASoC)

- Merged upstream in 2006
- Split driver framework into 3 categories of drivers
  - Platform: Copy data from memory to sound pipeline (DMA or PIO)
  - CPU DAI: Output data onto an external audio bus
  - CODEC: Mixing and ADC/DAC
- Fabric driver combines components
  - Describes external components (speaker, microphone)
ASoC – DAPM

- Dynamic Audio Power Management (DAPM)
- Graph-based description of functional modules
- Fine grained power tracking
- Cross device dependency management
Modern Hardware
High Definition Audio (HDA)

- Successor to AC'97
  - Introduced in 2004
- Follows the general approach of AC'97
  - Split between host controller and CODEC
  - But is a lot more flexible
- Self-describing
  - Allows generic driver
  - Quirks still required
High Definition Audio (HDA)

- Hierarchical function groups
- RPC style communication
- Used for all on-board audio
- Also used for HDMI and DisplayPort audio
Mobile

- Mobile devices have become the predominant multimedia devices
- Audio quality and features are a differentiating factor
  - Hardware is highly specialized
- Mobile distributions use a different development model
  - One software package per device
- Hardware specific sound servers
  - Aware of low level implementation details of the sound hardware
Present
Linux Consumer Audio Stack Today

Userspace:
- PulseAudio
- alsalib

Kernel:
- ALSA
- HDA driver
- USB driver

Hardware:
- HDA
- USB
Homogeneous Hardware Environment

- Virtually all shipped hardware had on-board HDA
- Most external hardware uses USB
- Large portion of driver development focuses on these platforms
- Small niche for professional and prosumer audio
  - Mostly Firewire based
Stable Software Environment

- **PulseAudio** is the default sound server on the majority of Linux distributions
  - Is aware of HDA and USB
  - Teething problems have been solved
- **Jack audio server** optionally available for more professional audio setups
  - Jack and PulseAudio know about each other and can negotiate hardware access
Embedded has Risen
## Driver Development Statistics

### Commits over the last 5 years

<table>
<thead>
<tr>
<th>Category</th>
<th>Commits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio total</td>
<td>14700</td>
</tr>
<tr>
<td>ALSA core</td>
<td>500</td>
</tr>
<tr>
<td>HDA</td>
<td>2100</td>
</tr>
<tr>
<td>Other PCI drivers</td>
<td>20-100 each (500 total)</td>
</tr>
<tr>
<td>USB</td>
<td>600</td>
</tr>
<tr>
<td>Firewire drivers</td>
<td>10-80 each (300 total)</td>
</tr>
<tr>
<td>ASoC</td>
<td>9500 (50/50 between CODEC and host)</td>
</tr>
</tbody>
</table>
Future
The next Transition
The next Transition

- Already started 1-2 years ago
- Concepts pioneered in the embedded/mobile sector are applied to all battery power hardware
- Devices are shipping now
Focus has shifted towards mobile multimedia
  - Power-limited due to battery operation
  - Power consumption has become a differentiating factor
Silicon has become much cheaper
Processing can be done more power efficient in specialized hardware
Keeping up
Use Case Manager (UCM)

- Groups control settings by function
  - E.g. “phone call”, “HiFi music”
- Sound servers select function
- Each unique component combination requires a set of UCM files
Topology

- Firmware file describing the hardware provided by userspace
- Initially intended to describe flow graph of a DSP firmware
- Now also used to describe hardware connections
Time for a major Overhaul?
Time for a Upgrade

- The hardware landscape has radically changed since the introduction of ALSA
- Core concepts of ALSA make for a very good, flexible and high-performance hardware abstraction layer
- Stay true to these core concepts
- Extend the framework to cover new use cases
New Component Model

- ASoC flattens component tree
  - Represented as single ALSA device
  - Applications can not identify which function belongs to which component

- Make components a top-level concept of ALSA
  - Allow applications to discover the hardware topology

- Most future sound cards will use ASoC
  - Make ASoC first level citizen in ALSA
New Component Model

- Concept of Platform, CPU-DAI, CODEC is overhauled
  - Functions are no longer clearly separated
  - Configuration space is shared between all components
- Remove distinction between different component types
- Introduce concept of domains and bridges
Applications do not know how a control affects the flow graph
  - Composite gain/attenuation
  - Routing

Export a annotated flow graph
Summary
Summary – Not a Happy End

• We are at the end of a golden era
• Hardware has already gone through the next transition
• Software has failed to keep up so far
• The next few years will be tough
  – There is hope though
Thanks
Q/A
Controls

• Controls are currently exported in a flattened hierarchy
  - Applications can not discover which control belongs where

• Create a component hierarchy which is used to group controls
Revamp ASoC Component Model

DMA
- Playback PCM domain
- Capture PCM domain

Audio controller
- Digital domain
- SRC
- Digital domain
- DAI

Codec
- Digital domain
- Link domain
- DAI
- DAC

Analog domain
- ADC